Angelo

Multiple Channels Real-Time Video
Frame Grabber Series
User's Manual

Recycled Paper

©Copyright 2003 ADLINK Technology Inc

All Rights Reserved.

Manual Rev 1.00: March 31, 2003

Part No: Part No: 50-12419-001

The information in this document is subject to change without prior notice in order to improve reliability, design and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

Trademarks

Angelo® is a registered trademark of ADLINK Technology Inc. Microsoft®, Windows NT®, Windows 98®, Windows 2000®, Windows XP® are registered trademarks of Microsoft Corporation. Borland C++ Builder® is a registered trademark of Borland International, Inc.

Other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Getting service from ADLINK

Customer Satisfaction is the most important priority for ADLINK Tech Inc. If you need any help or service, please contact us.

ADLINK Technology Inc.			
Web Site	http://www.adlink.com.tw http://www.adlinktechnology.com		
Sales & Service	service@adlink.com.tw		
Technical Support	NuDAQ	nudaq@adlink.com.tw	
	NuDAM	nudam@adlink.com.tw	
	NuIPC	nuipc@adlink.com.tw	
	NuPRO	nupro@adlink.com.tw	
	Software	sw@adlink.com.tw	
TEL	+886-2-82265877	FAX	+886-2-82265717
Address	9F, No. 166, Jian Yi Road, Chungho City, Taipei, 235 Taiwan.		

Please email or FAX us of your detailed information for a prompt, satisfactory and constant service.

Detailed Company Information			
Company/Organization			
Contact Person			
E-mail Address			
Address			
Country			
TEL		FAX	
Web Site			
Questions			
Product Model			
Environment to Use	OS: Computer Brand: M/B: CPU: Chipset: BIOS: Video Card: Network Interface Card: Other:		
Detail Description			
Suggestions for ADLINK			

Table of Contents

Introduction **1**
1.1 Features 2
1.1.1 Image Acquisition *2*
1.1.2 I/O Lines *2*
1.1.3 Watchdog Timer *3*
1.1.4 Supported software *3*
1.2 Applications 3

Hardware Reference **4**
2.1 RTV series 4
2.1.1 RTV24 Specification *4*
2.2 cRTV series 9
2.2.1 cRTV24 Specification *9*
2.2.2 cRTV44 Specification *11*
2.3 PMC-RTV series 16
2.3.1 PMC-RTV21 Specification *16*
2.3.2 PMC-RTV21G Specification *18*

Installation Guide **20**
3.1 Hardware Installation 20
3.1.1 RTV series *20*
3.1.2 cRTV Series *21*
3.1.3 PMC-RTV Series *22*

ViewCreator Utility **23**
4.1 Overview 23
4.2 Component description 24
4.3 Operation theory 25
4.3.1 Continuous grab *25*
4.3.2 Video image configuration *25*
4.3.3 Video adjustments *26*
4.3.4 Save image file *26*
4.3.5 Special image effect *27*
4.3.6 Tools *27*

Function Library **28**
5.1 List of Functions 28
5.2 C/C++ Programming Library 29
5.3 System Function 30
5.4 Configuration Function 33
5.5 Image Grabbing Function 38
5.6 GPIO & EEPROM Function 40
5.7 Callback & Thread Function 44
5.8 Watch Dog Timer 49
5.9 Software Trigger 51
5.10 Frame Buffer 54

Appendix: **58**
Appendix A: Architecture Example 58
Appendix B: Glossary 59

Warranty Policy **61**

1

Introduction

Angelo is a series acquisition board designed without compromise for security and video surveillance applications. It constitutes an ideal device for PC based multiple-channel digital video recorder.

This 32-bit/64bit, 33MHz/66MHz in PCI/cPCI/PMC bus frame grabber captures simultaneously four video analog streams in real time. It accepts standard composite color (PAL, NTSC) or monochrome video formats (CCIR, EIA).

The resolution is programmable including the square-pixel resolution (640 x 480 or 768 x 576) and the broadcast resolution. Before images are transferred into the PC' s memory, the image can be scaled down using selectable ratios.

Arbitrary cropping to a region of interest is possible. The RTV series generates bitmaps in all popular color formats such as RGB, YUV, planar or packed.

System integrators will benefit from a watchdog timer for fault-tolerant applications and from easy-to-use standard connectors.

1.1 Features

1.1.1 Image Acquisition

Acquisition speed

NTSC	1 Camera	2 Cameras	3 Cameras	4 Cameras	8 Cameras
Fields	60	120	180	240	240
Frames	30	60	90	120	120
PAL	1 Camera	2 Cameras	3 Cameras	4 Cameras	8 Cameras
Fields	50	100	150	200	200
Frames	25	50	75	100	100

Note: The PMC-RTV22 is capable of only up to 30 frames (60 fields) in total acquisition speed.

Color Image: The Color video format is compatible with the following composite video input formats: NTSC-M, NTSC-Japan, PCL-B, PAL-D, PAL-G, PAL-H, PAL-I, PAM-M, PAL-N and SECAM

Monochrome Image: The monochrome video acquisition is compatible with CCIR and EIA (RS-170)

Optional scaling: Optional scaling of acquired image or portions of an image.

- Acquisition of a programmable area of interest.
- Scaling of the image (down to 1:16).
- Adjustment of hue (for NTSC signals), contrast (0 to 200%), brightness and saturation (0 to 200% for U and V signals).
- Automatic chrominance gain control.

1.1.2 I/O Lines

The Angelo series are fitted with TTL compatible I/O lines protected against overloads and electrostatic discharges. Every line may be configured as an input or output. They can be used to trigger the acquisition or report alarms signal.

1.1.3 Watchdog Timer

A hardware watchdog is available on the RTV24. The watchdog is able to monitor the PC' s application operation and will automatically reset the PC after a programmable inactivity time-out. This ensures reliable operation of remote systems.

1.1.4 Supported software

WDM driver - The drivers support VC++/VB programming under Windows 98/2000/XP platform with DLL.

ViewCreator – The package will assist in initial test and functional evaluation.

1.2 Applications

- PC Based Surveillance System
- Digital Video Recorder (DVR)
- Factory Monitoring System
- Machine Vision Inspection System
- Scientific Research Instrumentation
- Medical Research Instrumentation

2

Hardware Reference

2.1 RTV series

2.1.1 RTV24 Specification

Video Input

- Four composite video color digitizers.
- Video input interface: Four composite BNC connectors.
- Co-axial cable suggested.

Channel Extension

- Up to 16 channels extension
- Channel extend interface
  - ✓ 10-pin ribbon cable to on-board 10-pin header connector for channel extension, each header can extend to 4 channels for video inputs.
  - ✓ Three 10-pin header connectors on-board.

General Purpose I/O Lines

- The I/O lines are TTL compatible and support 4 inputs, 4 outputs and 4 soft trigger lines.
- GPIO interface
  - ✓ Two 10-pin header connectors on-board.
  - ✓ The I/O lines are internally pulled up. They have the following characteristics:

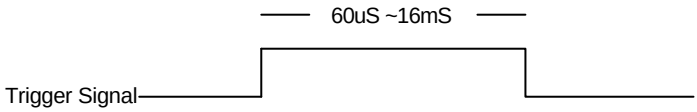
Voltage	MIN	MAX
Input high voltage (5ì A)	2.0V	5.25V
Input low voltage (-5ì A)	0.0V	0.80V
Output high voltage (-1.0mA)	5.0V	-
Output low voltage (100.0mA)	-	0.5V

Watch Dog Timer

- For monitoring the PC' s application operation and will reset the PC after a programmable inactivity time-out.
- Interface: 2-pin header

4-channel software trigger output

- 4 channels programmable trigger' s scale, scale from 60uS ~ 16mS



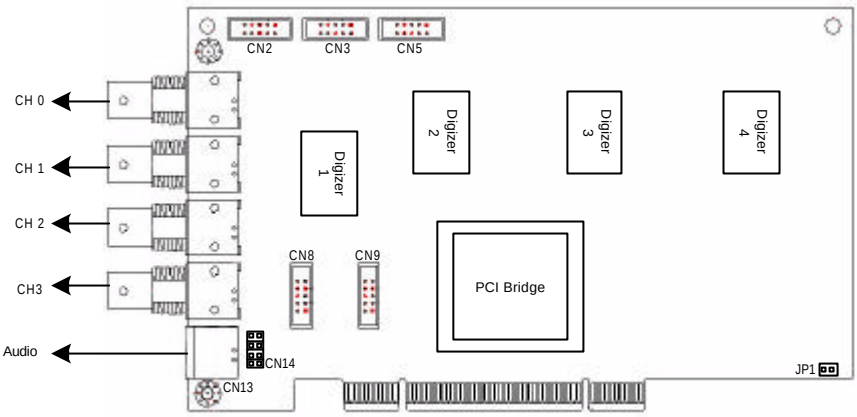
User EEPROM

- Support 1K bit EEPROM for user defined purposes

Form Factor

- 32bit/ 33/66MHz PCI bus half size board.

2.1.1.1 RTV24 PCB Outline Drawing

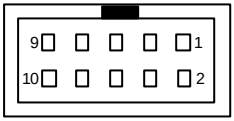


2.1.1.2 Connectors & Pin Definitions

Video Inputs

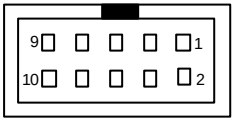
Connector	Definition
	Video IN – CH 0
	Video IN – CH 1
	Video IN – CH 2
	Video IN – CH 3

Channel Extension Video Input (CN2)



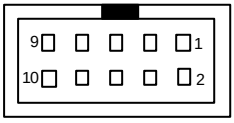
PIN NO.	Function	PIN NO.	Function
1	GND	2	CH4 video in
3	CH5 video in	4	GND
5	GND	6	CH6 video in
7	CH7 video in	8	GND
9	GND	10	GND

Channel Extension Video Input (CN3)



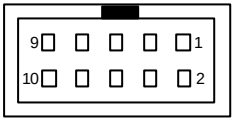
PIN NO.	Function	PIN NO.	Function
1	GND	2	CH8 video in
3	CH9 video in	4	GND
5	GND	6	CH10 video in
7	CH11 video in	8	GND
9	GND	10	GND

Channel Extension Video Input (CN5)



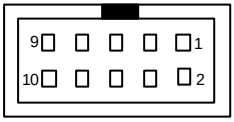
PIN NO.	Function	PIN NO.	Function
1	GND	2	CH12 video in
3	CH13 video in	4	GND
5	GND	6	CH14 video in
7	CH15 video in	8	GND
9	GND	10	GND

GPIO (CN8)



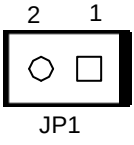
PIN NO.	Function	PIN NO.	Function
1	IN0 (External interrupt)	2	GND
3	OUT0	4	Software Trigger 0
5	IN1 (External interrupt)	6	Software Trigger 1
7	OUT1	8	+5V
9	GND	10	--

GPIO (CN9)



PIN NO.	Function	PIN NO.	Function
1	IN2 (External interrupt)	2	GND
3	OUT2	4	Software Trigger 2
5	IN3 (External interrupt)	6	Software Trigger 3
7	OUT3	8	+5V
9	GND	10	--

Watchdog Timer Reset



PIN NO.	Function
1	System reset
2	GND

2.2 cRTV series

2.2.1 cRTV24 Specification

Video Input

- Four composite video color digitizers.
- Video input interface: Four composite BNC connectors.
- LED for channel status report.
- Coaxial cable suggested.

Channel Extension

- Up to 8 channels extension
- Channel extend interface
  - ✓ 10-pin ribbon cable to on-board 10-pin header connector for channel extension, each header can extend to 4 channels for video inputs.

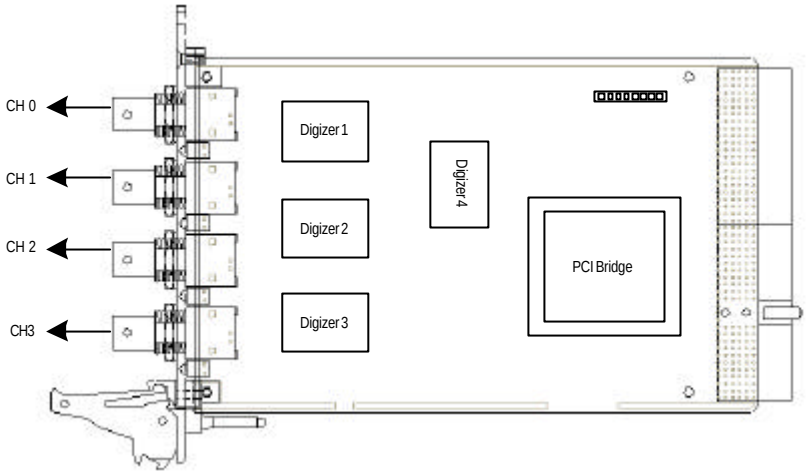
User EEPROM

- Support 1K bit EEPROM for user defined purposes

Form Factor

- 32/64bit/ 33/66MHz, 3U Compact PCI board.

2.2.1.1 cRTV24 PCB Outline Drawing

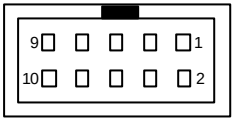


2.2.1.2 Connectors & Pin Definition

Video Inputs

Connector	Definition
	CH0 (Channel 0 BNC)
	CH0 (Status LED)
	CH1 (Channel 1 BNC)
	CH1 (Status LED)
	CH2 (Channel 2 BNC)
	CH2 (Status LED)
	CH3 (Channel 3 BNC)
	CH3 (Status LED)

Channel Extension Video Input (CN8)



PIN NO.	Function	PIN NO.	Function
1	GND	2	CH4 video in
3	CH5 video in	4	GND
5	GND	6	CH6 video in
7	CH7 video in	8	GND
9	GND	10	GND

2.2.2 cRTV44 Specification

Video Input

- Four composite video color digitizers.
- Video input interface: Four composite BNC connectors.
- LED for channel status report.
- Coaxial cable suggested.

General Purpose I/O Lines

- The I/O lines are TTL compatible and support 4 input, 4 output and 4 soft trigger lines.
- GPIO interface
  - ✓ Two 10-pin header connectors on-board.
  - ✓ The I/O lines are internally pulled up. They have the following characteristics:

Voltage	MIN	MAX
Input high voltage (20ì A)	2.0V	5.25V
Input low voltage (-0.2mA)	0.0V	0.80V
Output high voltage (-1.0mA)	5.0V	-
Output low voltage (100.0mA)	-	0.5V

Channel Extension

- Up to 8 channels extension
- Channel extend interface
  - ✓ 10-pin ribbon cable to on-board 10-pin header connector for channel extension, each header can extend to 4 channels for video inputs.

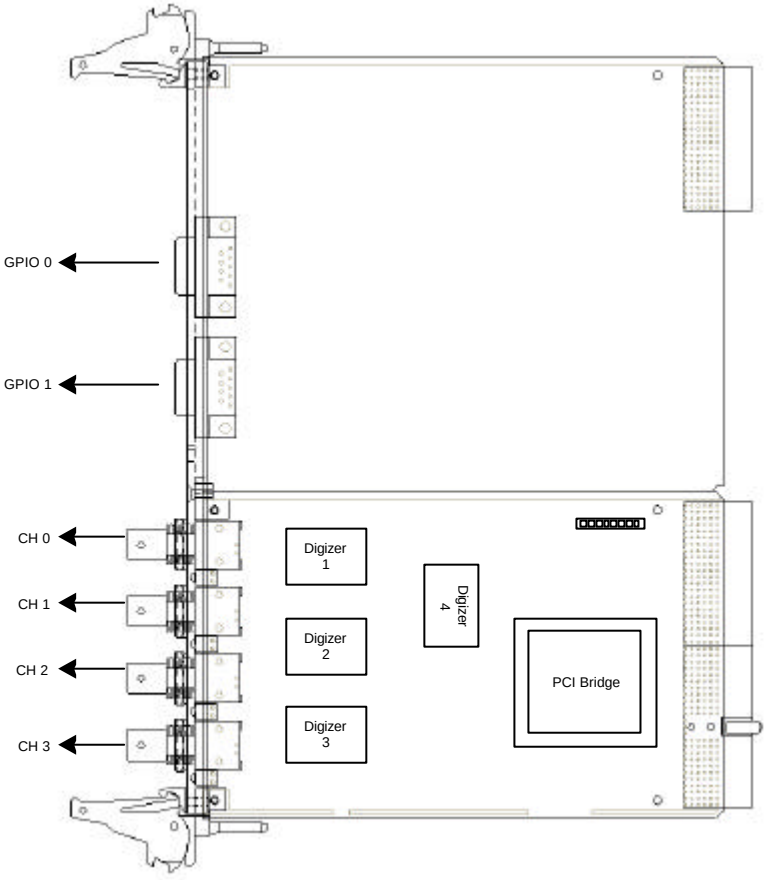
User EEPROM

- Support 1K bit EEPROM for user defined purposes

Form Factor

- 32/64bit/ 33/66MHz, 6U Compact PCI board.

2.2.2.1 cRTV44 PCB Outline Drawing

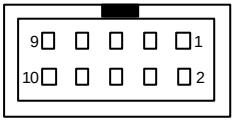


2.2.2.2 Connectors & Pin Definition

Video Input

Connector	Definition
	CH0 (Channel 0 BNC)
	CH0 (Status LED)
	CH1 (Channel 1 BNC)
	CH1 (Status LED)
	CH2 (Channel 2 BNC)
	CH2 (Status LED)
	CH3 (Channel 3 BNC)
	CH3 (Status LED)

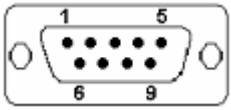
Channel Extension Video Input (CN8)



PIN NO.	Function	PIN NO.	Function
1	GND	2	CH4 video in
3	CH5 video in	4	GND
5	GND	6	CH6 video in
7	CH7 video in	8	GND
9	GND	10	GND

GPIO 0

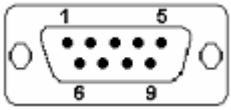
- PIN IN0 and OUT0 is used by channel 0
- PIN IN1 and OUT1 is used by channel 1



PIN NO.	Function	PIN NO.	Function
1	IN0 (External interrupt)	**6**	GND
2	OUT0	**7**	GND
3	IN1 (External interrupt)	**8**	GND
4	OUT1	**9**	+5V
5	GND		

GPIO 1

- PIN IN2 OUT2 is for channel 2
- PIN IN3 OUT3 is for channel 3



PIN NO.	Function	PIN NO.	Function
1	IN2 (External interrupt)	**6**	GND
2	OUT2	**7**	GND
3	IN3 (External interrupt)	**8**	GND
4	OUT3	**9**	+5V
5	GND		

2.3 PMC-RTV series

2.3.1 PMC-RTV21 Specification

Video Input

- Four composite video color digitizers.
- Video input interface: DB-9 female connectors.
- Coaxial cable suggested.

General Purpose I/O Lines

- The I/O lines are TTL compatible and support 1 input and 1 output.
- GPIO interface
  - ✓ One DB-9 male connector
  - ✓ The I/O lines are internally pulled up. They have the following characteristics:

Voltage	MIN	MAX
Input high voltage (20ì A)	2.0V	5.25V
Input low voltage (-0.2mA)	0.0V	0.80V
Output high voltage (-1.0mA)	5.0V	-
Output low voltage (100.0mA)	-	0.5V

User EEPROM

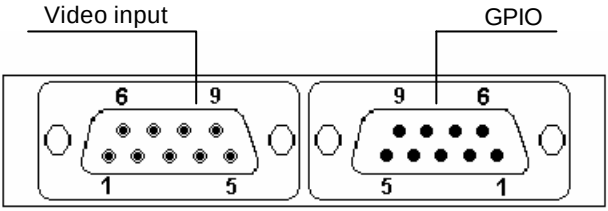
- Support 1K bit EEPROM for user defined purposes

Form Factor

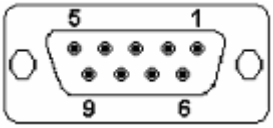
- 32bit/33MHz PMC socket board.

2.3.1.1 Connectors & Pin Definition

Video Input & GPIO

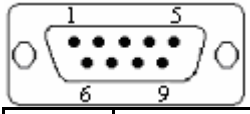


Video input



PIN NO.	Function	PIN NO.	Function
1	GND	**6**	CH0 video in
2	CH1 video in	**7**	GND
3	GND	**8**	CH2 video in
4	CH3 video in	**9**	GND
5	NC		

GPIO



PIN NO.	Function	PIN NO.	Function
1	IN0 (External interrupt)	**6**	GND
2	OUT0	**7**	GND
3	--	**8**	GND
4	--	**9**	+5V
5	GND		

2.3.2 PMC-RTV21G Specification

Video Input

- Four composite video color digitizers.
- Video input interface: 10-pin header connectors.
- Coaxial cable suggested.

General Purpose I/O Lines

- The I/O lines are TTL compatible and support 1 input and 1 output.
- GPIO interface
  - ✓ One 10-pin header connector.
  - ✓ The I/O lines are internally pulled up. They have the following characteristics:

Voltage	MIN	MAX
Input high voltage (20ì A)	2.0V	5.25V
Input low voltage (-0.2mA)	0.0V	0.80V
Output high voltage (-1.0mA)	5.0V	-
Output low voltage (100.0mA)	-	0.5V

User EEPROM

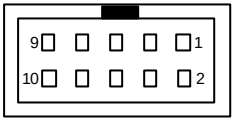
- Support 1K bit EEPROM for user defined purposes

Form Factor

- 32bit/33MHz PMC socket board.

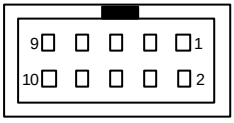
2.3.2.1 Connectors & Pin Definition

Video Input



PIN NO.	Function	PIN NO.	Function
1	GND	2	CH0 video in
3	CH1video in	4	GND
5	GND	6	CH2video in
7	CH3video in	8	GND
9	GND	10	GND

GPIO



PIN NO.	Function	PIN NO.	Function
1	IN0 External interrupt)	2	GND
3	OUT0	4	NC
5	NC	6	GND
7	NC	8	+5V
9	GND	10	NC

3

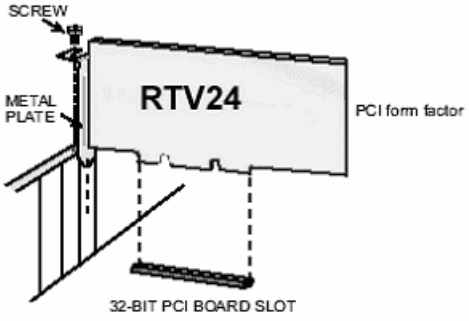
Installation Guide

3.1 Hardware Installation

3.1.1 RTV series

Use the following steps to install the RTV series board on the PCI bus

1. Remove the cover from your computer using the instructions from your computer manual
2. Check that you have an empty PCI (32-bit) slot that can accommodate the board. If you do not have an empty slot, remove a PCI board from your computer to make room for your RTV24 board and take note of the slot number you choose.
3. If present, remove the blank metal plate located at the back of the selected slot. Keep the removed screw; you will need it to fasten the RTV24 board after installation.
4. Carefully position the RTV24 in the selected PCI slot as illustrated below. If you are using a tower computer, orient the board to suit the board slots in your computer.



5. Once perfectly aligned with an empty slot, press the board firmly but carefully into the connector.
6. Anchor the board by replacing the screw that you removed.
7. Connect your video sources for image acquisition tests. For details, refer to the ' ViewCreator Utility" .
8. Turn on your computer. In some cases, when you boot up your computer, the " Plug-and-Play" feature of Windows will detect the new PCI card 8 times (4 videos and 4 audios) and you will be asked to assign a driver to it. For details, see the " Installation Guide" .

3.1.2 cRTV Series

Use the following steps to install the cRTV series board onto the Compact PCI bus

1. Remove the cover from your computer using the instructions from your computer manual
2. Check that you have an empty cPCI (32-bit/64-bit) slot that can accommodate the board. If you do not have an empty slot, remove a cPCI board from your computer to make room for your cRTV24 (3U) / cRTV44 (6U) board and take note of the slot number you choose.
3. If present, remove the blank metal plate located at the front of the selected slot. Keep the removed screw; you will need it to fasten the cRTV24 (3U) / cRTV44 (6U) board.
4. Carefully position the cRTV24 or cRTV44 in the selected cPCI slot as illustrated below.

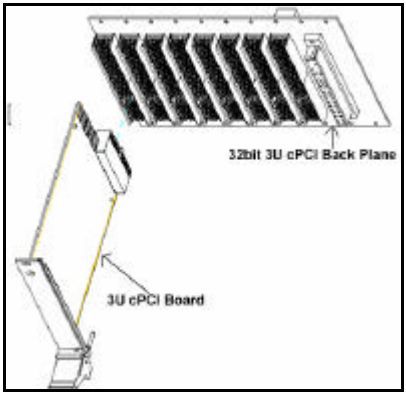


Figure 1. cRTV24 (3U cPCI)

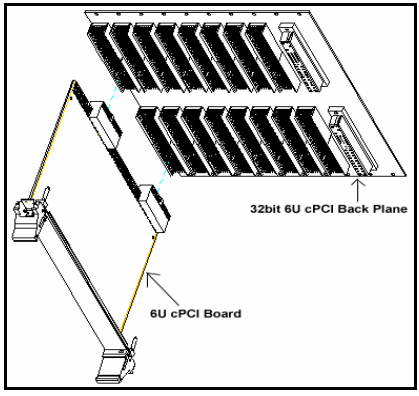


Figure 2. cRTV44 (6U cPCI)

5. Carefully slide the cRTV24 (3U)/cRTV44 (6U) along the guide of the chosen slot to the backplane and push the board firmly but carefully into the connector and then lock the board in place by pushing the release lever outwards.

6. Anchor the board by replacing the screw that you removed.

7. Connect your video sources for image acquisition tests. For details, refer to the ' ViewCreator Utility".

8. Turn on your computer. In some cases, when you boot up your computer, the "Plug-and-Play" feature of Windows will detect the new PCI card 8 times (4 videos and 4 audios) and you will be asked to assign a driver to it. For details, see the "Installation Guide".

3.1.3 PMC-RTV Series

The PMC socket maybe integrated with the cPCI CPU board or is a standalone system board for an embedded system. Use the following steps to install the PMC-RTV series board onto the PMC socket.

1. Remove the cover from your computer using the instructions from your computer manual

2. Check that you have an empty PMC (32-bit) socket that can accommodate the board. If you do not have an empty slot, remove a PMC board from your computer to make room for your PMC board.

3. Carefully position PMC-RTV24 onto the PMC socket.

4. Once perfectly aligned with an empty PMC socket, press the board firmly but carefully into the connector.

5. Connect your video sources for image acquisition tests. For details, refer to the ' ViewCreator Utility".

6. Turn on your computer. In some cases, when you boot up your computer, the "Plug-and-Play" feature of Windows will detect the new PCI card 8 times (4 videos and 4 audios) and you will be asked to assign a driver to it. For details, see the "Installation Guide".

4

ViewCreator Utility

After installing all the hardware, it is necessary to correctly configure the cards and double check that everything is ok before running an application. This chapter gives an outline for establishing a Vision system and manually exercising the Angelo series cards to verify correct operation. ViewCreator provides a simple yet powerful means to setup, configure, test and debug the Vision system with the Angelo series card in place.

Note: ViewCreator is available only for Windows 98/NT/2K/XP with screen resolution of higher than 800x600 and cannot run under DOS.

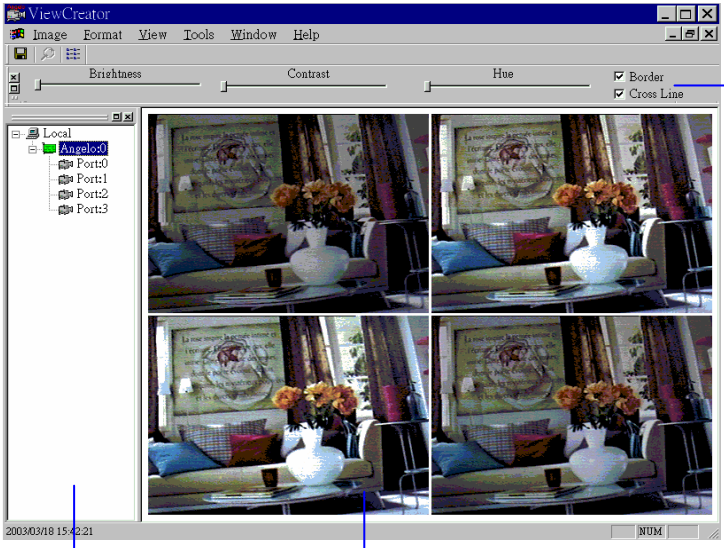
4.1 Overview

ViewCreator offers the following features and functionality:

1. 32-bit operation under Windows 98/NT/2K/XP
2. Access and configuration of Angelo series card
3. Ability to access all video picture adjustments
4. Ability to save image file into BMP or TIF
5. Direct access to the general purpose I/O
6. FULL, CIF, QCIF Image size, 2x2, 4x4 display
7. Software trigger ability

Note: ViewCreator is available only for Windows 98/NT/2K/XP with screen resolution of higher than 800x600 and cannot run under DOS.

4.2 Component description



Tree Browser

The Tree Browser window lists the Angelo series cards and video ports available at the local computer.

Image View

The Image View window displays the Full, CIF, QCIF size images and some image effect.

Control Panel

The control panel allows for making video adjustments including brightness, hue, contrast etc.

4.3 Operation theory

ViewCreator provides much functionality to access the Angelo series card. The functions are described below.

4.3.1 Continuous grab

Single channel display

- Click the "video port icon" in the Tree Browser window. A video frame will appear in the Image View window.

2x2 channels

- Click the "Angelo Series card icon" in the Tree Browser window. All video ports in this card will appear in the Image View window.

All channels

- Click the "Local icon" in the Tree Browser window. All video ports in the system will appear in the Image View window.

4.3.2 Video image configuration

Video format

- Click "Format" in the menu bar, and select the format of the video camera. The supported video formats are NTSC, EIA, PAL, and CCIR.

Color format

- The color format setting in ViewCreator is always RGB24 and is unchangeable but the color format of the application can changed.

Video size

- Click "View" in the menu bar and select the image size required. The supported video size listed below:
  - ✓ FULL: 640x480 for NTSC, EIA; 768x576 for PAL, CCIR
  - ✓ CIF: 320x240 for NTSC, EIA; 384x288 for PAL, CCIR
  - ✓ QCF: 160x120 for NTSC, EIA; 192x144 for PAL, CCIR

4.3.3 Video adjustments

Hue

- Click and hold down the left mouse button on the "Hue" slider of the Control Panel and drag the cursor to change the value. Values range from 0~255.

Contrast

- Click and hold down the left mouse button on the "Contrast" slider of the Control Panel and drag the cursor to change the value. Values range from 0~255

Brightness

- Click and hold down the left mouse button on the "Brightness" slider of the Control Panel and drag the cursor to change the value. Values range from 0~255

4.3.4 Save image file

This function can only take affect under single channel display mode. Selecting the video port in the Tree Browser window will cause ViewCreator to enter into the single channel display mode.

TIF

- Click "Image" in the menu bar and select "Save As" to bring up the *Save As* dialog box. Select the file location, TIF file format and enter the file name, then Click the OK button.

BMP

- Click "Image" in the menu bar and select "Save As" to bring up the *Save As* dialog box. Select the file location, BMP file format and enter the file name, then Click the OK button.

4.3.5 Special image effect

Border

- Check the "Border" check box in the Control Panel. A red dash borderline will appear around the image.
- Drag the red line to change the border. Only the image in the border gets refreshed.

Cross Line

- Check the "Cross Line" check box in the Control Panel. A cross line will appear in the center of the rectangle.

4.3.6 Tools

GPIO & LED

- Click "Tool" in the menu bar and select the "GPIO & LED" item to bring up the *GPIO dialog* box. Select the port you wish to access, and then select the digital output value. Click the write or read button to write/read to the digital I/O ports.
- LED status is only supported with the cPCI Angelo series card, set the LED status and check the LED on the Angelo series card.

EEPROM

- Click "Tool" in the menu bar and select the "EEPROM" item to bring up the EEPROM *dialog* box. Select the card you wish to access and enter the offset and output values, then click the write button to write the value into the EEPROM. Enter the offset value and click the read button to read the value from the EEPROM.
- Valid offset values is between 0~127. Valid output value is 0~255. The value in the EEPROM will not be erased when the system is powered off.

Software trigger

- Click "Tool" in the menu bar and select the "Software Trigger" item to bring up the Software Trigger *dialog* box. Select the card you wish to access, and then set the interval of your trigger pulse output. Check the ports you want to trigger simultaneously, and click the Trigger button.
- The pulse output is in a one shot format and the voltage goes high (from 0V to 5V).

5

Function Library

This chapter describes the supporting A.P.I. for the Angelo Series cards. User can use these functions to develop application programs under Visual C++, Visual Basic, C++ Builder and Delphi.

5.1 List of Functions

Category	Section	Function
System Function	5.3	AngeloRTV_Initial(PortNo)
		AngeloRTV_Close(PortNo)
		AngeloRTV_Software_Reset(PortNo)
Configuration Function	5.4	**AngeloRTV_Set_Image_Config(PortNo,ConfigIndex,Value)**
		AngeloRTV_Get_Image_Config(PortNo,ConfigIndex ,Value)
		AngeloRTV_Set_Color_Format(PortNo,ColorFormat)
		AngeloRTV_Get_Color_Format(PortNo,ColorFormat)
		AngeloRTV_Set_Video_Format(PortNo,Value)
		AngeloRTV_Get_Video_Format(PortNo, Value)
Image Grabbing Function	5.5	AngeloRTV_Capture_Start(PortNo, CaptureNo)
		AngeloRTV_Select_Channel(PortNo, Multiplex)
		AngeloRTV_Capture_Stop (PortNo)
GPIO & EPROM	5.6	AngeloRTV_Set_GPIO_Sts(PortNo, Status)
		AngeloRTV_Get_GPIO_Sts(PortNo, Status)
		AngeloRTV_Write_EEPROM(PortNo, Offset, Value)
		AngeloRTV_Read_EEPROM(PortNo, Offset, Value)
Callback & Thread	5.7	AngeloRTV_Get_Int_Status(PortNo,IntStatus)
		AngeloRTV_Set_Int_Factor(PortNo, IntFactor)
		AngeloRTV_Set_Int_Event(PortNo,hEvent)
		AngeloRTV_Set_Callback(PortNo, CallBackProc)
Watch Dog Timer	5.8	AngeloRTV_Set_WDT(CardNo, Enable, Interval)
Software Trigger	5.9	AngeloRTV_Trigger_Config(PortNo,Interval)
		AngeloRTV_Trigger_Start(CardNo, Multiplex)
Frame Buffer	5.10	AngeloRTV_Copy_frame(PortNo,Dest_Address,Size_Byte)
		AngeloRTV_Get_frame(PortNo,Start_Address, Size_Byte)
		AngeloRTV_Save_File(PortNo, FileName,FileFormat)

5.2 C/C++ Programming Library

This section gives details of programming codes. Function prototypes and common data types are defined in **Angelo.h**. The Angelo Series library uses these data types. We suggest that these data types be used in your application programs. The following table shows the data types and their range.

Type Name	Description	Range
U8	8-bit ASCII character	0 to 255
I16	16-bit signed integer	-32768 to 32767
U16	16-bit unsigned integer	0 to 65535
I32	32-bit signed long integer	-2147483648 to 2147483647
U32	32-bit unsigned long integer	0 to 4294967295
F32	32-bit single-precision floating-point	-3.402823E38 to 3.402823E38
F64	64-bit double-precision floating-point	-1.797683134862315E308 to 1.797683134862315E309
Boolean	Boolean logic value	TRUE, FALSE

5.3 System Function

@ Name

AngeloRTV_Initial(PortNo) – Initialize the port in Angelo series card.
AngeloRTV_Close(PortNo) – Close the port in Angelo series card.
AngeloRTV_Software_Reset(PortNo) – Reset the port in Angelo series card.

@ Description

AngeloRTV_Initial:

This function is used to initialize the ports of the Angelo Series card. Each application program must call this function before any other functions can be used. If the initialization is executed successfully, it returns a value of 0.

Note: There are four ports for the PCI-RTV-24, cPCI-RTV-24, cPCI-RTV-44 series card, and one port on the PMC-RTV-21.

AngeloRTV_Close:

This function is used to release all resource from the ports of the Angelo Series card.

AngeloRTV_Software_Reset:

This function is used to reset the port to its initial state.

@ Syntax

C/C++ (Windows 98/NT/2000/XP/CE.NET)

```
I16 AngeloRTV_Initial(U16 PortNo)
I16 AngeloRTV_Close(U16 PortNo)
I16 AngeloRTV_Software_Reset(U16 PortNo)
```

Visual Basic (Windows 98/NT/2000/XP/CE.NET)

```
AngeloRTV_Initial (ByVal PortNo As Integer) As Integer
AngeloRTV_Close(ByVal PortNo As Integer) As Integer
AngeloRTV_Software_Reset (ByVal PortNo As Integer) As Integer
```

Delphi (Windows 98/NT/2000/XP)

```
AngeloRTV_Initial(PortNo:Smallint):Smallint
AngeloRTV_ Close (PortNo:Smallint):Smallint
AngeloRTV_Software_Reset (PortNo:Smallint):Smallint
```

@ Argument

PortNo: Port number is the zero index of the Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, and each PCI-RTV-24 has four ports, so the first port of card 0 is "0", and the first port of card 1 is "4"

@ Return Code

0: ERROR_NoError

-2: ERROR_Card_Not_Exist – make sure the Angelo series card is plugged into your system, check the device manager to make sure the device is loaded, and the " PortNo" parameter is valid.

-3: ERROR_Card_Not_Accessible – make sure the Angelo series card is plugged into your system, check the device manager to make sure the device is loaded, and the " PortNo" parameter is valid.

-12: ERROR_CPLD_Check_Failed – Power off the computer and power on again.

@ Example

<VC/BCB >

AngeloRTV_Initial –

```
I16 Result;
for(int PortNo= 0 ; PortNo <4;PortNo++)
Result = AngeloRTV_Initial (PortNo);
```

AngeloRTV_Cose –

```
I16 Result;
for(int PortNo= 0 ; PortNo <4;PortNo++)
Result = AngeloRTV_Cose (PortNo);
```

AngeloRTV_Software_Reset–

```
I16 Result;
for(int PortNo= 0 ; PortNo <4;PortNo++)
Result = AngeloRTV_Software_Reset (PortNo);
```

< Visual Basic >

AngeloRTV_Initial –

```
Dim Result As Integer
Dim PortNo As Integer
For PortNo= 0 To 3
Result = AngeloRTV_Initial (ByVal PortNo)
```

AngeloRTV_Cose –

```
Dim Result As Integer
Dim PortNo As Integer
For PortNo= 0 To 3
Result = AngeloRTV_Close (ByVal PortNo)
```

AngeloRTV_Software_Reset–

```
Dim Result As Integer
Dim PortNo As Integer
For PortNo= 0 To 3
Result = AngeloRTV_Software_Reset(ByVal PortNo)
```

<Delphi >

AngeloRTV_Initial –

```
var PortNo,Result:SmallInt;
   for i:= 0 to 3 do
     begin
     Result := AngeloRTV_Initial (PortNo);
     End;
```

AngeloRTV_Cose –

```
var PortNo,Result:SmallInt;
   for i:= 0 to 3 do
     begin
     Result := AngeloRTV_Close (PortNo);
     End;
```

AngeloRTV_Software_Reset–

```
var PortNo,Result:SmallInt;
   for i:= 0 to 3 do
     begin
     Result := AngeloRTV_Software_Reset (PortNo);
     End;
```

5.4 Configuration Function

@ Name

AngeloRTV_Set_Image_Config(PortNo,ConfigIndex ,Value) – Set the video adjustments.
AngeloRTV_Get_Image_Config(PortNo,ConfigIndex ,Value) –Get the video adjustments.
AngeloRTV_Set_Color_Format(PortNo,ColorFormat) – Set the color format.
AngeloRTV_Get_Color_Format(PortNo,ColorFormat) – Get the color format.
AngeloRTV_Set_Video_Format(PortNo,Value) – Set the video format.
AngeloRTV_Get_Video_Format(PortNo, Value) – Set the video format.

@ Description

AngeloRTV_Set_Image_Config:

This function is used to adjust the hue, contrast, Saturation and brightness of the port for the Angelo series card

AngeloRTV_Get_Image_Config:

This function is used to retrieve the current hue, contrast, Saturation and brightness setting of the port for the Angelo series card

AngeloRTV_Set_ Color _Format:

This function is used to set the color format of the port for the Angelo series card, the valid color format are gray scale, RGB, YUV.

AngeloRTV_Get_ Color _Format:

This function is used to retrieve the color format of the port for the Angelo series card

AngeloRTV_Set_ Video_Format:

This function is used to set the Video format of the port for the Angelo series card, the valid color formats are NTSC, EIA, PAL, CCIR

AngeloRTV_Get_ Video _Format:

This function is used to retrieve the Video format of the port for the Angelo series card

@ Syntax

C/C++ (Windows 98/NT/2000/XP/CE.NET)

```
I16 AngeloRTV_Set_Image_Config(U16 PortNo,U8 ConfigIndex ,U8 Value);
I16 AngeloRTV_Get_Image_Config(U16 PortNo,U8 ConfigIndex ,U8* Value);
I16 AngeloRTV_Set_ Color _Format (U16 PortNo, U8 ColorFormat);
I16 AngeloRTV_Get_ Color _Format (U16 PortNo, U8* ColorFormat);
I16 AngeloRTV_Set_Video _Format (U16 PortNo,U8 VideoFormat);
I16 AngeloRTV_Set_Video _Format (U16 PortNo, U8* VideoFormat);
```

Visual Basic (Windows 98/NT/2000/XP/CE.NET)

AngeloRTV_Set_Image_Config(ByVal PortNo As Integer, ByVal ConfigIndex As Byte,ByVal Value As Byte) As Integer
AngeloRTV_Get_Image_Config(ByVal PortNo As Integer, ByVal ConfigIndex As Byte, ByRef Value As Byte) As Integer
AngeloRTV_Set_ Color _Format (ByVal PortNo As Integer, ByVal ColorFormat As Byte) As Integer
AngeloRTV_Get_ Color _Format (ByVal PortNo As Integer, ByRef ColorFormat As Byte) As Integer
AngeloRTV_Set_Video _Format (ByVal PortNo As Integer, ByVal VideoFormat As Byte) As Integer
AngeloRTV_Set_Video _Format (ByVal PortNo As Integer, ByRef VideoFormat As Byte) As Integer

Delphi (Windows 98/NT/2000/XP)

AngeloRTV_Set_Image_Config(PortNo:Smallint;ConfigIndex:Byte;Value:Byte): Smallint; AngeloRTV_Get_Image_Config(PortNo:Smallint;ConfigIndex:Byte;var Value:Byte):Smallint;
AngeloRTV_Set_Color_Format(PortNo:Smallint;ColorFormat:Byte):Smallint;
AngeloRTV_Get_Color_Format(PortNo:Smallint;var ColorFormat:Byte):Smallint;
AngeloRTV_Set_Video_Format(PortNo:Smallint;VideoFormat:Byte):Smallint;
AngeloRTV_Get_Video_Format(PortNo:Smallint;var VideoFormat:Byte):Smallint;

@ Argument

PortNo: Port number is the zero index of the Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, and each PCI-RTV-24 has four ports, so the first port of card 0 is "0", and the first port of card 1 is "4"

ConfigIndex:

0 for BRIGHTNESS
1 for HUE
2 for SATURATION(U)
3 for SATURATION(V)
4 for CONTRAST (LUMA)
5 for luma notch filter(for monochrome video, the notch filter should not be used)

Value: (**0~255**)

	Range	Default value
BRIGHTNESS	0 ---- 255	128
HUE	0 ---- 255	0
CHROMA(U)	0 ---- 255	127
CHROMA(V)	0 ---- 255	127
LUMA	0 ---- 255	108
luma notch filter	---- 0(Enable) or 1(Disable)	

Color Format:

RGB16 = 0,
GRAY = 1,
RGB15 = 2,
RGB24 = 3,
RGB32 = 4,
RGB8 = 5,
RAW8X = 6,
YUY24:2:2= 7,
BtYUV 4:1:1= 8,

Video Format:

Full NTSC(640*480) = 0,
Full PAL(768*576) = 1,
CIF NTSC(320*240) = 2,
CIF PAL(384*288) = 3,
QCIF NTSC(160*120) = 4,
QCIF PAL(192*144) = 5,

@ Return Code

0: ERROR_NoError

-7: ERROR_Not_Initialized – Make sure the port has been initialized by "AngeloRTV_Initial".

-9: ERROR_Invalid_PortNo – Please input the correct "PortNo" parameter.

-10: ERROR_Invaild_Video_Format – the valid video format should be between 0 and 5.

-11: ERROR_Invalid_Color_Format– the valid video format should be between 0 and 8.

@ Example

<VC/BCB >

AngeloRTV_Set_Image_Config –

AngeloRTV_Get_Image_Config –

```
I16 Result;
I16 PortNo = 0;
U8 ConfigIndex = 0;
U8 Value = 128;
Result = AngeloRTV_Set_Image_Config (PortNo,
   ConfigIndex, Value);
Result = AngeloRTV_Get_Image_Config (PortNo,
   ConfigIndex, &Value);
```

AngeloRTV_Set_Color_Format –

AngeloRTV_Get_Color_Format –

AngeloRTV_Set_Video_Format –

AngeloRTV_Get_Video_Format –

```
I16 Result;
I16 PortNo = 0;
U8 VideoFormat = 0;
U8 ColorFormat = 3;
Result = AngeloRTV_Set_Color_Format(PortNo,
   ColorFormat);
Result = AngeloRTV_Get_Color_Format(PortNo, &
   ColorFormat);
Result = AngeloRTV_Set_Video_Format(PortNo,
   VideoFormat);
Result = AngeloRTV_Get_Video_Format(PortNo,
   &VideoFormat);
```

< Visual Basic >

AngeloRTV_Set_Image_Config –

AngeloRTV_Get_Image_Config –

```
Dim Result As Integer
Dim PortNo As Integer
Dim ConfigIndex As Byte
Dim Value As Byte
PortNo = 0
ConfigIndex =0
Value = 128
Result = AngeloRTV_Set_Image_Config (ByVal PortNo, ByVal
   ConfigIndex, ByVal Value)
Result = AngeloRTV_Get_Image_Config (ByVal PortNo, ByVal
   ConfigIndex, ByRef Value)
```

AngeloRTV_Set_Color_Format –

AngeloRTV_Get_Color_Format –

AngeloRTV_Set_Video_Format –

AngeloRTV_Get_Video_Format –

```
Dim Result As Integer
Dim PortNo As Integer
Dim ColorFormat As Byte
Dim VideoFormat As Byte
PortNo = 0
ColorFormat =3
VideoFormat = 0

Result = AngeloRTV_Set_Color_Format(ByVal PortNo, ByVal
   ColorFormat)
Result = AngeloRTV_Get_Color_Format(ByVal PortNo, ByRef
   ColorFormat)
Result = AngeloRTV_Set_Video_Format(ByVal PortNo, ByVal
   VideoFormat)

  Result = AngeloRTV_Get_Video_Format(ByVal PortNo, ByRef
VideoFormat)
```

<Delphi >

AngeloRTV_Set_Image_Config –

AngeloRTV_Get_Image_Config –

```
Var
Result : SmallInt;
PortNo : SmallInt;
ConfigIndex: Byte;
Value: Byte;
PortNo:=0;
ConfigIndex:=0;
Value:=0;
Result := AngeloRTV_Set_Image_Config (PortNo,
   ConfigIndex, Value);
Result := AngeloRTV_Get_Image_Config (PortNo,
   ConfigIndex, Value);
```

AngeloRTV_Set_Color_Format –

AngeloRTV_Get_Color_Format –

AngeloRTV_Set_Video_Format –

AngeloRTV_Get_Video_Format –

```
Var
Result : SmallInt;
PortNo : SmallInt;
VideoFormat: Byte;
ColorFormat: Byte;
PortNo:=0;
VideoFormat:=0;
ColorFormat:=3;
Result := AngeloRTV_Set_Color_Format(PortNo,
   ColorFormat);
Result := AngeloRTV_Get_Color_Format(PortNo,
   ColorFormat);
Result := AngeloRTV_Set_Video_Format(PortNo,
   VideoFormat);
     Result := AngeloRTV_Get_Video_Format(PortNo, VideoFormat);
```

5.5 Image Grabbing Function

@ Name

AngeloRTV_Capture_Start(PortNo, CaptureNo) – Start to grab the video image
AngeloRTV_Select_Channel(PortNo, Multiplex) – channel extension of video signal, for advanced only
AngeloRTV_Capture_Stop(PortNo) –Stop to grab the video image

@ Description

AngeloRTV_Capture_Start :

This function is used to continuously grab video frames and stops when the total frame number equals the " CaptureNo" parameter, the frame update rate is 30 frames/sec. If the " CaptureNo" is 0xFFFFFFFF, the frame grabbing won't stop until you call the " AngeloRTV_Capture_Stop" function.

AngeloRTV_Capture_Stop :

This function is used to stop grabbing video frames.

AngeloRTV_Select_Channel :

Angelo Series card is capable of channel extension; this function is used to multiplex the video signal for the ports. In most case usage this function should not be required because the default setting is one port is dedicated to one channel.

Note: Do not use this function call if you don't have the channel extension board

@ Syntax

C/C++ (Windows 98/NT/2000/XP/CE.NET)

```
I16 AngeloRTV_Capture_Start (U16 PortNo, U32 CaptureNo)
I16 AngeloRTV_Select_Channel (U16 PortNo, U16 Multiplex)
I16 AngeloRTV_Capture_Stop (U16 PortNo)
```

Visual Basic (Windows 98/NT/2000/XP/CE.NET)

AngeloRTV_Capture_Start (ByVal PortNo As Integer, ByVal CaptureNo As Long) As Integer

AngeloRTV_Select_Channel (ByVal PortNo As Integer, ByVal Multiplex As Integer) As Integer

AngeloRTV_Capture_Stop (ByVak PortNo As Integer) As Integer

Delphi (Windows 98/NT/2000/XP)

AngeloRTV_Capture_Start (PortNo:Smallint; CaptureNo:LongInt):Smallint

AngeloRTV_Select_Channel (PortNo:Smallint; Multiplex:SmallInt):Smallint

AngeloRTV_Capture_Stop (PortNo:Smallint):Smallint

@ Argument

PortNo: Port number is the zero index of the Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, and each PCI-RTV-24 has four ports, so the first port of card 0 is "0", and the first port of card 1 is "4"

CaptureNo: Capture number is the total frames to grab, if this parameter is 0xFFFFFFFF, the grabbing won't stop until "AngeloRTV_Capture_Stop" is issued.

Multiplex: Indicates the multiplex channels, the bit definition is as follows:

```
Bit 0 : Channel 0, 0 for disable ; 1 for enable.
Bit 1 : Channel 1, 0 for disable ; 1 for enable.
Bit 2 : Channel 2, 0 for disable ; 1 for enable.
Bit 3 : Channel 3, 0 for disable ; 1 for enable.
For example:
Multiplex = 1, only channel 0 is enable
Multiplex = 2, only channel 1 is enable
Multiplex = 15, four channels are enable
```

@ Return Code

0: ERROR_NoError

-7: ERROR_Not_Initialized – Make sure the port has been initialized by "AngeloRTV_Initial".

-9: ERROR_Invalid_PortNo – Please input the correct "PortNo" parameter.

@ Example

<VC/BCB >

AngeloRTV_Capture_Start –

AngeloRTV_Capture_Stop –

```
I16 Result;
U16 PortNo = 0;
U32 CaptureNo = 0xFFFFFFFF;
Result = AngeloRTV_Capture_Start (PortNo, CaptureNo);
Result = AngeloRTV_Capture_Stop (PortNo);
```

< Visual Basic >

AngeloRTV_Capture_Start –

AngeloRTV_Capture_Stop –

```
Dim Result As Integer
Dim PortNo As Integer
Dim CaptureNo As Long
PortNo = 0
CaptureNo = &HFFFFFFFF
Result = AngeloRTV_Capture_Start (ByVal PortNo, ByVal
   CaptureNo)
     Result = AngeloRTV_Capture_Stop (ByVal PortNo)
```

<Delphi >

```
Var
Result : SmallInt;
PortNo: SmallInt;
CaptureNo: LontInt;
PortNo:=0;
CaptureNo:= INFINITE;
Result := AngeloRTV_Capture_Start (PortNo, CaptureNo);
     Result: = AngeloRTV_Capture_Stop (PortNo);
```

5.6 GPIO & EEPROM Function

@ Name

AngeloRTV_Set_GPIO_Sts (PortNo, Status) – Set Digital Output status.
AngeloRTV_Get_GPIO_Sts (PortNo, Status) –Get Digital Input status.
AngeloRTV_Write _EEPROM (PortNo, Offset, Value) – Write data into EEPROM
AngeloRTV_Read_EEPROM (PortNo, Offset, Value) – Read data from EEPROM

@ Description

AngeloRTV_Set_GPIO_Sts:

There is one digital output channel in each port of the Angelo series card, use this function to set the digital output status.

AngeloRTV_Get_GPIO_Sts:

There is one digital input channel in each port of Angelo series card, use this function to get the digital input status.

AngeloRTV_Write _EEPROM:

This function is used towrite data into the EEPROM, the data in EEPROM will not be lost even when powered off.

AngeloRTV_Read_EEPROM:

This function is used to read data from the EEPROM, the data in EEPROM will not be lost evenwhen powered off.

@ Syntax

C/C++ (Windows 98/NT/2000/XP/CE.NET)

```
I16 AngeloRTV_Set_GPIO_Sts(U16 PortNo,U8 Status);
I16 AngeloRTV_Get_GPIO_Sts(U16 PortNo,U8 * Status);
I16 AngeloRTV_Write_EEPROM(U16 CardNo, U8 Offset, U8 Value);
I16 AngeloRTV_Read_EEPROM(U16 CardNo, U8 Offset, U8* Value);
```

Visual Basic (Windows 98/NT/2000/XP/CE.NET)

```
AngeloRTV_Set_GPIO_Sts (ByVal PortNo As Integer, ByVal Status As Byte) As
 Integer
AngeloRTV_Get_GPIO_Sts (ByVal PortNo As Integer, ByRef Status As
Byte) As Integer
AngeloRTV_Write_EEPROM (ByVal PortNo As Integer, ByVal Offset As Byte,
 ByVal Value As Byte) As Integer
AngeloRTV_Read_EEPROM (ByVal PortNo As Integer, ByVal Offset As Byte,
 ByRef Value As Byte) As Integer
```

Delphi (Windows 98/NT/2000/XP)

```
AngeloRTV_Set_GPIO_Sts (PortNo:Smallint;status:Byte):Smallint;
AngeloRTV_Get_GPIO_Sts (PortNo:Smallint;var status:Byte):Smallint;
AngeloRTV_Write_EEPROM( PortNo:Smallint;Offset:Byte;Value:Byte):Smallint;
AngeloRTV_Read_EEPROM( PortNo:Smallint; Offset:Byte;var
 Value:Byte):Smallint;
```

@ Argument

PortNo: Port number is the zero index of the Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, and each PCI-RTV-24 has four ports, so the first port of card 0 is "0", and the first port of card 1 is "4"

Status: The digital input or digital output status
0 Low
1 High

Offset: The offset address of the EEPROM, this parameter is valid between 0 and 127

Value: The value in Byte data type, this parameter is valid between 0 and 255

@ Return Code

0: ERROR_NoError

-7: ERROR_Not_Initialized – Make sure the port has been initialized by "AngeloRTV_Initial".

-9: ERROR_Invalid_PortNo – Please input the correct "PortNo" parameter.

-15: ERROR_Invalid_Address – the valid offset address should be between 0 and 127

@ Example

<VC/BCB >

AngeloRTV_Set_GPIO_Sts –

AngeloRTV_Get_GPIO_Sts –

```
I16 Result;
I16 PortNo = 0;
U8 Status = 1;
Result = AngeloRTV_Set_GPIO_Sts (PortNo, Status);
Result = AngeloRTV_Get_GPIO_Sts (PortNo, & Status);
```

AngeloRTV_Write_EEPROM

AngeloRTV_Read_EEPROM

```
I16 Result;
I16 PortNo = 0;
U8 Offset = 0;
U8 Value = 128;
Result = AngeloRTV_Write_EEPROM (PortNo, Offset, Value);
Result = AngeloRTV_Read_EEPROM (PortNo, Offset, &Value);
```

< Visual Basic >

AngeloRTV_Set_GPIO_Sts –

AngeloRTV_Get_GPIO_Sts –

```
Dim Result As Integer
Dim PortNo As Integer
Dim Status As Byte
PortNo = 0
Status = 1
Result = AngeloRTV_Set_GPIO_Sts (ByVal PortNo, ByVal
  Status)
Result = AngeloRTV_Get_GPIO_Sts (ByVal PortNo, ByRef
  Status)
```

AngeloRTV_Write_EEPROM

AngeloRTV_Read_EEPROM

```
Dim Result As Integer
Dim PortNo As Integer
Dim Offset As Byte
Dim Value As Byte
PortNo = 0
Offset =0
Value = 128
```

Result = ***AngeloRTV_Write_EEPROM****(ByVal PortNo, ByVal Offset, ByVal Value)*

Result = ***AngeloRTV_Read_EEPROM****(ByVal PortNo, ByVal Offset, ByRef Value)*

<Delphi >

AngeloRTV_Set_GPIO_Sts –

AngeloRTV_Get_GPIO_Sts –

```
Var
Result : SmallInt;
PortNo : SmallInt;
Status: Byte;
PortNo:=0;
Status:=1;
Result := AngeloRTV_Set_GPIO_Sts (PortNo, Status);
Result := AngeloRTV_Get_GPIO_Sts (PortNo, Status);
```

AngeloRTV_Write_EEPROM

AngeloRTV_Read_EEPROM

```
Var
Result : SmallInt;
PortNo : SmallInt;
Offset: Byte;
Value: Byte;
PortNo:=0;
Offset:=0;
Value:=128;
Result := AngeloRTV_Write_EEPROM (PortNo, Offset,
  Value);
Result := AngeloRTV_Read_EEPROM (PortNo, Offset, Value);
```

5.7 Callback & Thread Function

@ Name

AngeloRTV_Get_Int_Status (PortNo, IntStatus) – Get the current interrupt status.
AngeloRTV_Set_Int_Factor((PortNo, IntFactor) –Set the factor to initiate the interrupt
AngeloRTV_Set_Int_Event (PortNo,hEvent) – assign the windows interrupt event
AngeloRTV_Set_Callback(PortNo, CallBackProc) – set the callback function when interrupt generates

@ Description

AngeloRTV_Get_Int_Status:

This function allows user to identify what cause the interrupt signal, the definition for each bit is as following.

Bit 0:GPIO interrupt, when Digital input channel is changed.
Bit 1:Channel 0 Image ready
Bit 2:Channel 1 Image ready
Bit 3:Channel 2 Image ready
Bit 4:Channel 3 Image ready

Note: There are four channels in each port, the default channel is channel 0

AngeloRTV_Set_Int_Factor:

This function allows user to select factors to initiate the event interrupt, the definition for each bit is as following.

Bit 0:GPIO interrupt
Bit 1:Image ready interrupt

AngeloRTV_Set_Int_Event:

This function is used to link the interrupt event, user only have to declare the "hEvent" variable, and call this function to DLL, the DLL will link the event and interrupt automatically.

AngeloRTV_Set_Callback:

This function is used to link the callback function when interrupt generates to host pc

Note: There are two ways to use the synchronization mechanism, one is the callback function, and the other is the thread function.

@ Syntax

C/C++ (Windows 98/NT/2000/XP/CE.NET)

```
I16 AngeloRTV_Get_Int_Status(U16 PortNo,U32 *IntStatus);
I16 AngeloRTV_Set_Int_Factor(U16 PortNo,U32 IntFactor);
I16 AngeloRTV_Set_Int_Event(U16 PortNo,HANDLE* hEvent);
I16 AngeloRTV_Set_Callback (U16 PortNo, void ( __stdcall *CallBackProc)(U32
 VideoBufferaddress ,U16 PortNo));
```

Visual Basic (Windows 98/NT/2000/XP/CE.NET)

AngeloRTV_Set_Int_Event(ByVal PortNo As Integer, ByRef hEvent As Long) As Integer

AngeloRTV_Set_Int_Factor(ByVal PortNo As Integer, ByVal IntFactor As Long) As Integer

AngeloRTV_Get_Int_Status(ByVal PortNo As Integer, ByRef IntStatus As Long) As Integer

AngeloRTV_Set_Callback(ByVal PortNo As Integer, ByVal CallBack As Long) As Integer

Delphi (Windows 98/NT/2000/XP)

AngeloRTV_Set_Int_Event(PortNo:Smallint;var hEvent:Integer):Smallint;

AngeloRTV_Set_Int_Factor(PortNo:Smallint;IntFactor:Longint):Smallint;

AngeloRTV_Get_Int_Status(PortNo:Smallint;var IntStatus:Longint):Smallint;

@ Argument

PortNo: Port number is the zero index of the Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, and each PCI-RTV-24 has four ports, so the first port of card 0 is "0", and the first port of card 1 is "4"

IntStatus: Interrupt status

Bit 0:GPIO interrupt, when Digital input channel is changed.
Bit 1:Channel 0 Image ready
Bit 2:Channel 1 Image ready
Bit 3:Channel 2 Image ready
Bit 4:Channel 3 Image ready

IntFactor: Interrupt flags

Bit 0:GPIO interrupt, 0 : for disable; 1 : for enable.
Bit 1:Image ready interrupt, 0 : for disable; 1 : for enable.

hEvent: Interrupt event handle.

@ Return Code

0: ERROR_NoError

-7: ERROR_Not_Initialized – Make sure the port has been initialized by "AngeloRTV_Initial".

-9: ERROR_Invalid_PortNo – Please input the correct "PortNo" parameter.

@ Example

<VC/BCB >

Use Thread:

```
HANDLE hEvent=NULL;
  void *pThread=NULL;
  U32 threadID;
  U16 PortNo = 0;
  U32 IntFactor = 0x03; //GPIO&Image Ready
  DWORD       nObj;
  U32 Size_Byte;
  U32         Status =0;
  I16 ISR_ON=0;
    DWORD WINAPI IntThreadProc( LPVOID lpParam )
        {
          while( ISR_ON )
          {
          nObj = WaitForSingleObject(hEvent, INFINITE);
          AngeloRTV_Get_Int_Status(PortNo,&Status);
          if((Status&0x01)==1)//GPIO
          {
          }
          if((Status>>1&0x01)==1)//Channel 0 of the nPort
          {
          }
          else if((Status>>2&0x01)==1)//Channel 1 of the nPort
          {
          }
          else if((Status>>3&0x01)==1)//Channel 2 of the nPort
          {
          }
          else if((Status>>4&0x01)==1)//Channel 3 of the nPort
          {
          }
          ResetEvent(hEvent);
          }

                Return TRUE;
        }
```

*AngeloRTV_Set_Int_Factor**(PortNo, IntFactor);***

*AngeloRTV_Set_Int_Event**(PortNo,&hEvent);***

pThread =CreateThread(NULL, 0, IntThreadProc, 0, 0, &threadID);

Use Callback Function:

```
U16 PortNo = 0;
void __stdcall MediaStreamProc( U32
   VideoBufferaddress ,U16 PortNo)
{
   U32 Status;
   AngeloRTV_Get_Int_Status(PortNo,&Status);
   if((Status&0x01)==1)//GPIO
   {
   }
   if((Status>>1&0x01)==1)//Channel 0 of the nPort
   {
   }
   else if((Status>>2&0x01)==1)//Channel 1 of the nPort
   {
   }
   else if((Status>>3&0x01)==1)//Channel 2 of the nPort
   {
   }
   else if((Status>>4&0x01)==1)//Channel 3 of the nPort
   {
   }
 }
AngeloRTV_Set_Callback (Port No,MediaStreamProc);
```

< Visual Basic >

Use Callback Function

```
    Dim Result As Integer
    Dim PortNo As Integer
Public Sub lpcallback(ByVal VideoBufferaddress As Long,
   ByVal PortNo As Integer)

   Dim Status As Long
   Result  = AngeloRTV_Get_Int_Status(PortNo, Status)

End Sub
PortNo = 0
Result = AngeloRTV_Set_Callback(PortNo, AddressOf
   lpcallback)
```

<Delphi >

Use Thread

```
Var
ISR_ON : SmallInt;
Event_Angelo:Integer;
ThreadId : LongInt;
PortNo: SmallInt;
PortNo:=0;
 function ThreadFunc(Parameter: Pointer): Integer ;
 var
   Str_Add :Pointer;
   Size_Byte :Longint;
   intstatus : LongInt;
begin
   while(ISR_ON=1) do
   begin
           WaitForSingleObject(Event_Angelo,INFINITE);
           ResetEvent(Event_Angelo);
           AngeloRTV_Get_Int_Status(PortNo,intstatus);
           if intstatus =2 then //image ready for channel
 0 of port
           begin

           end;
   end;
end;
AngeloRTV_Set_Int_Factor(PortNo,3);//GPIO & Image Ready
  AngeloRTV_Set_Int_Event(PortNo,Event_Angelo);
 ISR_ON :=1;
  Mythread :=
 BeginThread(nil,0,ThreadFunc,nil,0,ThreadId);
```

Use Callback function

```
 var
PortNo: SmallInt;
PortNo:=0;
 procedure MyCallback(VideoBufferAddress :
 LongInt;PortNo : SmallInt);stdcall
var
    Str_Add :Pointer;
    Result :Smallint;
    Size_Byte :LongInt;
    intstatus :LongInt;
begin

   AngeloRTV_Get_Int_Status(PortNo,intstatus);
   if intstatus = 2 then
   begin
   end;
 end;
AngeloRTV_Set_Callback(Cur_Port,MyCallback);
```

5.8 Watch Dog Timer

@ Name

AngeloRTV_Set_WDT(CardNo, Enable, Interval) – Set the watch dog status

@ Description

AngeloRTV_Set_WDT :

This function is used to enable or disable the watchdog timer for the Angelo series card, and set the interval of the timer. When the user has enabled the watchdog timer and selected the 16 seconds interval, the reset signal will be triggered if this function is not called within 16 seconds periodically.

@ Syntax

C/C++ (Windows 98/NT/2000/XP/CE.NET)

I16 AngeloRTV_Set_WDT (U16 CardNo,U16 Enable,U16 Interval)

Visual Basic (Windows 98/NT/2000/XP/CE.NET)

AngeloRTV_Set_WDT (ByVal PortNo As Integer, ByVal Enable As Integer, ByVal Interval As Integer) As Integer

Delphi (Windows 98/NT/2000/XP)

AngeloRTV_Set_WDT(CardNo:Smallint;enable:Smallint;interval:Smallint):Smallint;

@ Argument

CardNo: Card number is the zero index for Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, so the "CardNo" of card 0 is 0, and 4 for card 1.

Enable: enable or disable the watchdog timer, 0 for disable, 1 for enable.

Interval: Indicates the watch dog timer interval, the definition is as following

1 : 8 seconds
2 : 16 seconds
3 : 32 seconds

@ Return Code

0: ERROR_NoError

-7: ERROR_Not_Initialized – Make sure the port has been initialized by "AngeloRTV_Initial".

-9: ERROR_Invalid_PortNo – Please input the correct "PortNo" parameter.

@ Example

<VC/BCB >

AngeloRTV_Set_WDT

```
I16 Result;
U16 CardNo = 0;
U16 Enable = 1;
U16 Interval = 1;
Result = AngeloRTV_Set_WDT(CardNo,Enable,Interval);
```

< Visual Basic >

AngeloRTV_Set_WDT

```
Dim Result As Integer
Dim CardNo As Integer
Dim Enable As Integer
Dim Interval As Integer
CardNo = 0
Enable = 1
Interval = 1
    Result = AngeloRTV_Set_WDT(CardNo,Enable,Interval)
```

<Delphi >

AngeloRTV_Set_WDT

```
Var
Result : SmallInt;
CardNo: SmallInt;
Enable: SmallInt;
Interval: SmallInt;
CardNo :=0;
Enable:=1;
Interval:=1;
Result := AngeloRTV_Set_WDT(CardNo,Enable,Interval);
```

5.9 Software Trigger

@ Name

AngeloRTV_Trigger_Config (PortNo,Interval) – Set software trigger configuration
AngeloRTV_Trigger_Start (CardNo, Multiplex) – generate single or multiple trigger output simultaneity

@ Description

AngeloRTV_Trigger_Config :

This function is used to configure the pulse output interval.

AngeloRTV_Trigger_Start :

This function is used to generate one shot pulse output for single or multiple ports.

@ Syntax

C/C++ (Windows 98/NT/2000/XP/CE.NET)

```
I16 AngeloRTV_Trigger_Config(U16 PortNo,U16 Interval);
I16 AngeloRTV_Trigger_Start(U16 CardNo,U16 Multiplex);
```

Visual Basic (Windows 98/NT/2000/XP/CE.NET)

```
AngeloRTV_Trigger_Config (ByVal PortNo As Integer, ByVal Interval As Integer)
 As Integer
AngeloRTV_Trigger_Start (ByVal CardNo As Integer, ByVal Multiplex As Integer)
 As Integer
```

Delphi (Windows 98/NT/2000/XP)

```
AngeloRTV_Trigger_Config (PortNo:Smallint; Interval:Smallint):Smallint;
AngeloRTV_Trigger_Start (CardNo:Smallint; Multiplex:Smallint):Smallint;
```

@ Argument

CardNo: Card number is the zero index for the Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, so the "CardNo" of card 0 is 0, and 1 for card 1.

PortNo: Port number is the zero index in Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, and each PCI-RTV-24 has four ports, so the first port of card 0 is"0", and the first port of card 1 is"4"

Interval: Indicates the trigger output interval, the valid range is from 0 to 253, the definition is as following

0 : 16 ms
32 : 12 ms
128 : 8 ms
253 : 60 µs

Multiplex: Indicates the trigger output ports in Angelo series card, the bit definition is as following

```
Bit 0 : Port 0 in each card, 0 for disable ; 1 for enable.
Bit 1 : Port 1 in each card, 0 for disable ; 1 for enable.
Bit 2 : Port 2 in each card, 0 for disable ; 1 for enable.
Bit 3 : Port 3 in each card, 0 for disable ; 1 for enable.
```

For example:

```
Multiplex = 1, only port 0 in Angelo series card generates trigger
output.
Multiplex = 2, only port 1 in Angelo series card generates trigger
output.
Multiplex = 15, four ports in Angelo series card generate trigger
output.
```

@ Return Code

0: ERROR_NoError

-7: ERROR_Not_Initialized – Make sure the port has been initialized by "AngeloRTV_Initial".

-9: ERROR_Invalid_PortNo – Please input the correct "PortNo" parameter.

@ Example

<VC/BCB >

AngeloRTV_Trigger_Config

AngeloRTV_Trigger_Start

```
I16 Result;
U16 CardNo = 0;
U16 PortNo = 0;
U16 Multiplex = 1;
U16 Interval = 32;
Result = AngeloRTV_Trigger_Config(PortNo,Interval);
Result = AngeloRTV_Trigger_Start (CardNo, Multiplex);
```

< Visual Basic >

AngeloRTV_Trigger_Config

AngeloRTV_Trigger_Start

```
Dim Result As Integer
Dim CardNo As Integer
Dim PortNo As Integer
Dim Multiplex As Integer
Dim Interval As Integer
CardNo = 0
PortNo = 0
Multiplex = 1
Interval = 32
    Result = AngeloRTV_Trigger_Config (PortNo,Interval)
    Result = AngeloRTV_Trigger_Start (CardNo, Multiplex)
```

<Delphi >

AngeloRTV_Trigger_Config

AngeloRTV_Trigger_Start

```
Var
Result : SmallInt;
CardNo: SmallInt;
PortNo: SmallInt;
Multiplex: SmallInt;
Interval: SmallInt;
CardNo :=0;
PortNo:=0;
Multiplex:=1;
Interval:=32;
Result := AngeloRTV_Trigger_Config (PortNo,Interval);
Result := AngeloRTV_Trigger_Start (CardNo, Multiplex);
```

5.10 Frame Buffer

@ Name

AngeloRTV_Copy_frame (PortNo, Dest_Address, Size_Byte) – Copy the frame date to the user allocated destination memory in size of byte.
AngeloRTV_Get_frame(PortNo, Start_Address, Size_Byte) – Get the frame memory start address and size of frame in byte.
AngeloRTV_Save_File(PortNo, FileName, FileFormat) – Save the video frame into image file.

@ Description

AngeloRTV_Copy_frame :

This function is used to copy the frame data to the memory or array that the user has allocated. Before using this function, remember to allocate enough memory address space or array elements.

AngeloRTV_Get_frame:

This function is used to retrieve the memory start address from the frame data, and the size in byte of the frame data, for example a FULL NTSC RGB24 video frame will occupy 900K Byte (640*480*3) memory address space.

AngeloRTV_Save_File :

This function is used to save the current video frame into image file, the supported file format are BMP and TIF.

@ Syntax

C/C++ (Windows 98/NT/2000/XP/CE.NET)

```
I16 AngeloRTV_Copy_Frame(U16 PortNo,U8 *Dest_Address,U32 Size_Byte);
I16 AngeloRTV_Get_Frame(U16 PortNo,U32* Start_Address,U32* Size_Byte);
16 AngeloRTV_Save_File(U16 PortNo, char* FileName,U8 FileFormat);
```

Visual Basic (Windows 98/NT/2000/XP/CE.NET)

```
AngeloRTV_Copy_Frame (ByVal PortNo As Integer, Dest_Address As Byte,
 ByVal Size_byte As Long) As Integer
AngeloRTV_Get_Frame (ByVal PortNo As Integer, Start_Address As Long,
 Size_byte As Long) As Integer
AngeloRTV_Save_File (ByVal PortNo As Integer, ByVal FileName As String,
 ByVal FileFormat As Byte) As Integer
```

Delphi (Windows 98/NT/2000/XP)

```
AngeloRTV_Copy_Frame(PortNo:Smallint;var
 Dest_Address:Byte;Size_byte:Longint):Smallint;
AngeloRTV_Get_Frame(PortNo:Smallint;var Start_Address:Pointer;var
 Size_byte:Longint):Smallint;
AngeloRTV_Save_File(PortNo:Smallint;FileName:String;FileFormat:Byte):Smalli
 nt;
```

@ Argument

PortNo: Port number is the zero index of the Angelo series card, for example, there are two PCI-RTV-24 Angelo cards (card 0, card 1) in your system, and each PCI-RTV-24 has four ports, so the first port of card 0 is"0", and the first port of card 1 is"4"

Dest_Address: The destination memory address or array that user allocated,

Start_Address: The memory start address of the video frame.

Size_Byte: The memory size in byte.

FileName: The file name user want to save, remember to add the file extension name.

FileFormat: The file format user want to save into, the definition is as following.

```
0: TIF image format
1: BMP image format
```

@ Return Code

0: ERROR_NoError

-7: ERROR_Not_Initialized – Make sure the port has been initialized by "AngeloRTV_Initial".

-9: ERROR_Invalid_PortNo – Please input the correct "PortNo" parameter.

@ Example

<VC/BCB >

AngeloRTV_Copy_Frame

```
I16 Result;
U16 PortNo = 0;
U32 Size_Byte = 640*480*3;
U8* Dest_Address =NULL;
Dest_Address = (U8*)malloc(Size_Byte );
Result = AngeloRTV_Copy_Frame (PortNo, Dest_Address,
Size_Byte);
```

AngeloRTV_Get_Frame

```
I16 Result;
U16 PortNo = 0;
U32 Size_Byte ;
U32 StrAddr ;
Result =
   AngeloRTV_Get_Frame(PortNo,&StrAddr,&Size_Byte);
```

AngeloRTV_Save_File

```
I16 Result;
U16 PortNo = 0;
Result = AngeloRTV_Save_File(PortNo,"Image.tif",0);
```

< Visual Basic >

AngeloRTV_Copy_Frame

```
        Dim Result As Integer
        Dim PortNo As Integer
        Dim Size_Byte As Long
        Dest_Address( ) As Byte
        PortNo = 0
        Size_Byte =640*480*3
        ReDim Dest_Address(0 To Size_Byte - 1) As Byte
        Result = AngeloRTV_Copy_Frame (PortNo, Dest_Address(0),
Size_Byte);
```

AngeloRTV_Get_Frame

```
        Dim Result As Integer
        Dim PortNo As Integer
 Dim Size_Byte As Long
 Dim StrAddr As Long
 PortNo = 0
        Result = AngeloRTV_Get_Frame( ByVal PortNo, Str_Add,
Size_Byte)
```

AngeloRTV_Save_File

```
        Dim Result As Integer
 PortNo = 0
        Result = AngeloRTV_Save_File (PortNo, " Image.tif" , 0)
```

<Delphi >

AngeloRTV_Copy_Frame

```
 Var
 Result : SmallInt;
 PortNo: SmallInt;
 Size_Byte :Longint;
 Dest_Add : array of Byte;
 PortNo := 0;
 Size_Byte := 640*480*3;
 SetLength(Dest_Add, Size_Byte);
        Result := AngeloRTV_Copy_Frame (PortNo, Dest_Add[0],
Size_Byte);
```

AngeloRTV_Get_Frame

```
Var
Result : SmallInt;
PortNo: SmallInt;
Size_Byte :Longint;
Str_Add :Pointer;
PortNo:=0;
Result :=AngeloRTV_Get_Frame(PortNo, Str_Add,
   Size_Byte);
```

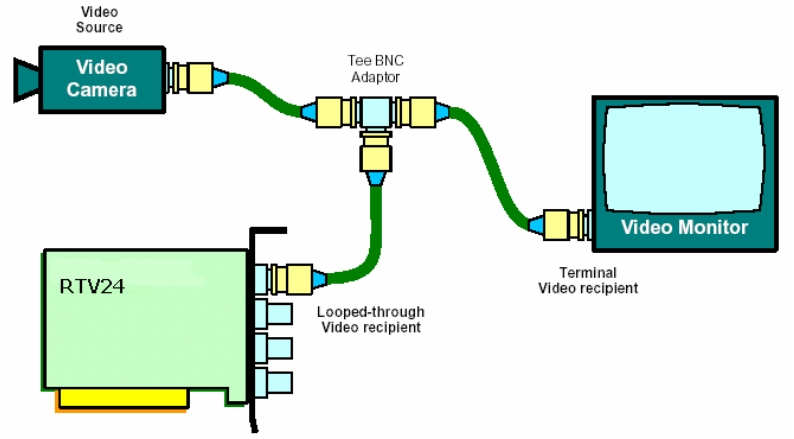
AngeloRTV_Save_File

```
Var
Result : SmallInt;
     PortNo: SmallInt;
PortNo:=0;
Result := AngeloRTV_Save_File (PortNo, 'Image.tif', 0)
```

A

Appendix:

Appendix A: Architecture Example



Appendix B: Glossary

Brightness:

Attribute of a visual sensation according to which an area appears to exhibit more or less light

CCIR:

A shortcut to designate a scanning system used in Europe. The CCIR system is made of two interlaced field of 312.5 lines, for a total of 625 lines. In each field, only 287.5 lines are conveying visible lines, for a total of 575 visible lines. A line lasts 64 ms, of which approximately 52 ms are conveying visible pixels.

Composite Video:

Composite video (CVS/CVBS) signal carries video picture information for color, brightness and synchronizing signals for both horizontal and vertical scans.

CIF:

CIF has 352(H) x 288(V) luminance pixels, and 176(H) x 144(V) chrominance pixels. QCIF is a similar picture format with one-quarter the size of CIF

EIA:

A shortcut to designate a scanning system used in America and Japan. The EIA system is made of two interlaced field of 262.5 lines, for a total of 525 lines. In each field, only 242.5 lines are conveying visible lines, for a total of 485 visible lines (typical value). A line lasts 63.56 ms, of which approximately 52 ms are conveying visible pixels.

Field:

For interlaced video the total picture is divided into two fields, one even and one odd each contain one half of the total vertical information. Each field takes one sixtieth of a second (one fiftieth for PAL) to complete. Two fields make a complete frame of video.

Frame:

One frame (two fields) of video contains the full vertical interlaced information content of the picture. For NTSC this consists of 525 lines and PAL a frame is consisted of 625 lines.

Gamma:

Cathode ray tubes (CRTs) do not have a linear relationship between brightness and the input voltage applied. To compensate for this non-linearity, a pre distortion or gamma correction is applied, generally at the camera source. A value of gamma equal to 2.2 is typical, but can very for different CRT phosphors.

Hue:

Attribute of a visual sensation according to which an area appears to be similar to one, or proportions of two, of the perceived colors red, yellow, green, and blue.

NTSC:

Shortcut to designate a color television broadcast standard used in America and Japan. The (M) NTSC system uses 525 lines per frame (2 interlaced fields), a 29.97 frame per second update rate, and the YIQ or YUV color space. In each field, only 242.5 lines are conveying visible lines, for a total of 485 visible lines (typical value). A line lasts 63.56 ms, of which approximately 52 ms are conveying visible pixels.

PAL:

Shortcut to designate a color television broadcast standard used in Europe. The (B, G, H, I) PAL (or Phase Alternation Line) uses 625 lines per frame (2 interlaced fields), a 25 frame per second update rate, and the YUV color space. In each field, only 287.5 lines are conveying visible lines, for a total of 575 visible lines. A line lasts 64 ms, of which approximately 52 ms are conveying visible pixels.

Saturation:

A characteristic describing color amplitude or intensity. A color of a given hue may consist of low or high saturation value, which relates to the vividness of color.

Warranty Policy

Thank you for choosing ADLINK. To understand your rights and enjoy all the after-sales services we offer, please read the following carefully.

1. Before using ADLINK's products, please read the user manual and follow the instructions exactly. When sending in damaged products for repair, please attach an RMA application form.

2. All ADLINK products come with a two-year guarantee, free of repair charge.

   - The warranty period starts from the product's shipment date from ADLINK's factory
   - Peripherals and third-party products not manufactured by ADLINK will be covered by the original manufacturers' warranty
   - End users requiring maintenance services should contact their local dealers. Local warranty conditions will depend on the local dealers

3. Our repair service does not cover two-year guarantee while damages are cause by the following:

   a. Damage caused by not following instructions on user menus.

   b. Damage caused by carelessness on the users' part during product transportation.

   c. Damage caused by fire, earthquakes, floods, lightening, pollution and incorrect usage of voltage transformers.

   d. Damage caused by unsuitable storage environments with high temperatures, high humidity or volatile chemicals.

   e. Damage caused by leakage of battery fluid when changing batteries.

   f. Damages from improper repair by unauthorized technicians.

   g. Products with altered and damaged serial numbers are not entitled to our service.

   h. Other categories not protected under our guarantees.

4. Customers are responsible for the fees regarding transportation of damaged products to our company or to the sales office.

5. To ensure the speed and quality of product repair, please download an RMA application form from our company website www.adlinktech.com. Damaged products with RMA forms attached receive priority.

For further questions, please contact our FAE staff.

ADLINK: service@adlinktech.com

Test & Measurement Product Segment: NuDAQ@adlinktech.com

Automation Product Segment: Automation@adlinktech.com
Computer & Communication Product Segment: NuPRO@adlinktech.com; NuIPC@adlinktech.com